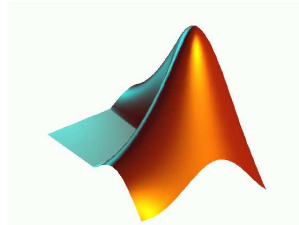


LINEAR ALGEBRA



MATLAB

Team: _____

Problem I: Solving linear systems

Code: I_solving_poisson.m

1. Getting started.

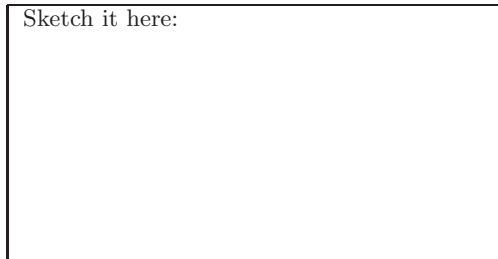
Run the provided code. Observe that it has the following form:

- A square matrix $A \in \mathbb{R}^{N \times N}$, $N = n^2$, with a special structure is formed (so-called *Poisson matrix*).
- A right-hand-side b with a special structure is formed (so-called *forcing*).
- There are several different ways to solve the system $Ax = b$ with built-in MATLAB routines which can be uncommented.
- The solution x is a vector containing the solution to the so-called (discrete) *Poisson equation*. Reshaping x into a matrix $XX \in \mathbb{R}^{n \times n}$ and plotting the entries of XX as a surface gives (an approximation) to the *Poisson equation*.
- NOTE: You do not have to know anything about physics to be able to work on this exercise. You will learn a bit about the equation by playing with the system parameters, though.

2. Matrix Structure.

- Look up what the commands `eye`, `diag`, `spy` do by typing `help eye`, etc. into the command window.
- Change the parameter `n` to $2^3, 2^4$ and use `spy(A)` to understand the structure of A .

Sketch it here:



3. Solving via rref.

- Uncomment the part where $Ax = b$ is solved by row reduction and uncomment the plotting part at the very bottom. Comment the `spy(A)` command.
- Notice that increasing the parameter `n` makes the execution time longer (see what the commands `tic`, `toc` display in the command window). Report them here for `n= 2^2, 2^3, 2^4`:

-
- Change the plot style from `surf` into `mesh` into `contour` into `imagesc`. Attach your favorite plot.

4. Manipulating the right-hand-side b .

Change b to understand its effect on the solution.

- Change the sign of `force`. What happens?
- Change the magnitude of `force` (from -1 to -10 to -100). What happens?
- Try the following combinations with fixed `n= 2^4`:

force	point_force	describe its effect
0	-100	
10	-100	
0	0	

- What everyday phenomenon do the plots remind you of?

5. **Solving via inversion: *Fill-in* for sparse matrices.**

Fix `force=0` and `point_force=-10`. Comment the part where the system is solved by row reduction and uncomment the part where it is solved by inversion. Run the program again for $n=2^2, 2^3, 2^4$ and record the times here:

Using `spy`, try to find out if A^{-1} is sparse. Google what *fill-in* for matrices means. How is it related to the problem here?

6. **Speed up through sparsity of A .**

Uncomment the line `A=sparse(A)` and notice the speed up for $n=2^3$ and $n=2^4$. Try to understand what this command does and describe the cause of the speed up briefly:

7. **Solving linear systems through factorization: The command `x=A\b`.**

Comment the part where the system is solved by inversion and uncomment the part where `x=A\b` is used. Run the program again for $n=2^2, 2^3, 2^4$, and notice the speed up by recording the times

and comparing to the previous times. Lookup in the online documentation of MATLAB what the command `x=A\b` does and give a brief explanation. Did we discuss anything related in this course?

8. **Testing the limits.**

Try to crank up n to test the limits of your computer (careful, it might crash at some point). How high did you get? $n =$ _____

9. **Result gallery.**

Play around with the parameters as you wish and attach your 3 favorite plots with information on all the parameters you chose and the solution method.

Problem II: Span of vectors, solution set of linear systems and orthogonality

Code: II_span.m

1. Getting started.

Run the program and try to understand what it does. Change the vectors v_1, v_2, p , change the limits and step size of the weight vectors c_1, c_2 . Change the color and style of the plot. Rotate the plot.

- Attach your favorite plot specifying the v_1, v_2, p you used.

2. Span and solution set of linear systems.

- Solve $2x_1 + 2x_2 - 4x_3 = 0$ by hand and write the solution set in parametric form. Use the provided code to plot the solution set.

$$x_h = \underline{\hspace{15em}}$$

- Solve $2x_1 + 2x_2 - 4x_3 = 16$ by hand and write the solution set in parametric form. Use the provided code to plot the solution set.

$$x = \underline{\hspace{15em}}$$

3. Orthogonality and inner product $u^T v$.

Two vectors $u, v \in \mathbb{R}^n$ are orthogonal if

$$u^T v = u_1 v_1 + \dots + u_n v_n = 0.$$

This type of product is so important, it has a name: $u^T v$ is called the *inner product of u and v* . Try the following example:

- Compute the inner product of $u = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$ and $v = \begin{bmatrix} -1 \\ 1 \end{bmatrix}$.

$$u^T v = \underline{\hspace{15em}}.$$

- What does this tell you about the angle between them? Sketch u and v .

4. Orthogonality, inner product, cross product and normal vector of a plane.

Take a look at the equation you just solved, namely, $2x_1 + 2x_2 - 4x_3 = 0$. It can be written as

$$n^T x = 0, \quad n = \begin{bmatrix} 2 \\ 2 \\ -4 \end{bmatrix}, \quad x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}.$$

The vector n is called *normal vector* of the plane that is the solution set of this equation. Include n in the previous plot and attach it. For v_1, v_2 a basis of the solution set, compute the inner products

$$n^T v_1 = \underline{\hspace{15em}},$$

$$n^T v_2 = \underline{\hspace{15em}}.$$

- What is the angle between n and any vector in the plane? $\underline{\hspace{5em}}$
- Use the command `cross(v_1, v_2)`. What does this command? How is the result related to the plane?

Problem III: Fitting data, least-squares and the space of polynomials

Code: III_fitting_data.m

1. **Getting started.**

Run the program and try to understand what it does.

2. **Fitting data.**

Assume you are given data points as, for instance, $\mathbf{t}_k, \mathbf{b}_k$ in the code. By mere inspection of the graph of the data you might have a first idea of what the curve b that fits these data points is. You might guess, for instance,

$$b(t) = x_3 t^2 + x_2 t + x_1, \quad x_1, x_2, x_3 \in \mathbb{R} \text{ unknown coefficients,}$$

so b is in the space of polynomials of degree 2 or less. One can use linear algebra techniques to determine x_1, x_2, x_3 .

- Uncomment the fitting part of the code and run it. Try to understand what it does.
- Try to extract from the provided code the procedure and briefly describe it here.

3. **Modifying the first guess.**

Assume you guess instead that

$$b(t) = x_4 t^3 + x_3 t^2 + x_2 t + x_1, \quad x_1, x_2, x_3, x_4 \in \mathbb{R} \text{ unknown coefficients.}$$

Modify the code to determine the unknown coefficients.

$$x_1 = \underline{\hspace{2cm}}, x_2 = \underline{\hspace{2cm}}, x_3 = \underline{\hspace{2cm}}, x_4 = \underline{\hspace{2cm}}.$$

4. **Changing the data.**

Change some numbers in the data and see how the fitting is adjusting. Attach your 2 favorite plots.

5. **Improving the guess.**

Did our second guess really improve anything? What would have been a smarter improvement?

Problem IV: Eigenvalues and parameters

Code: IV_eigenvalues_and_parameters.m

1. **Getting started.**

Run the program and try to understand what it does.

2. **Parameters and diagonalization.**

Find $a \in \mathbb{R}$ such that A is diagonalizable.

$a =$ _____.

Explain how diagonalization could fail.

3. **Parameters and diagonalization in 3 dimensions.**

Modify the code for the matrix

$$A = \begin{bmatrix} -1 & 1 & 1 \\ 0 & \cos(a) & 1 \\ 0 & 0 & \sin(a) \end{bmatrix}.$$

Plot all eigenvalue curves into the same plot with different styles and attach the plot.

Problem V: Difference equations and how eigenvalues predict the future

Code: V_difference_equations.m

1. Getting started.

Run the program and try to understand what it does.

2. Eigenvalues and long-term behavior of the sequence $x_{k+1} = Ax_k$.

Try the following values for the parameter m . Report what you see by sketching the plots qualitatively.

m	eigenvalues of A	describe what you observe
-0.5		
-0.4375		
-0.4		
0.1		

Make a conjecture about the relation between the behavior of the sequence and the eigenvalues of A .

3. Designing a difference equation with prescribed behavior.

Design a matrix A such that it has the following eigenvalues and eigenvectors.

$$\lambda_1 = -0.5, v_1 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \quad \lambda_2 = 2, v_2 = \begin{bmatrix} -1 \\ 2 \end{bmatrix}.$$

Hint: Use the diagonalization formula!

$$A = \underline{\hspace{10em}}.$$

Sketch what you expect for the initial values

$$[17, 17], [-0.25, 0.5], [1, -2], [17, 18],$$

and confirm your sketch by modifying and running the code. Attach the corresponding plot.